

**Sistema de Evacuação**

2MIEIC02 – Grupo 3 – Tema 3

**Elaborado por:**

Luís Borges – up201605859

Patrícia Janeiro – up[201605946](https://sigarra.up.pt/feup/pt/fest_geral.cursos_list?pv_num_unico=201605946)

Sandro Campos – up[201605947](https://sigarra.up.pt/feup/pt/fest_geral.cursos_list?pv_num_unico=201605947)

**Índice**

[1. Introdução 3](#_Toc511227295)

[2. Descrição do Problema 4](#_Toc511227296)

[Input 4](#_Toc511227297)

[Introdução de dados 4](#_Toc511227298)

[Output 4](#_Toc511227299)

[Objetivo 4](#_Toc511227300)

[Restrições 4](#_Toc511227301)

[3. Algoritmos implementados 5](#_Toc511227302)

[4. Casos de Utilização 6](#_Toc511227303)

[5. Diagrama de Classes 9](#_Toc511227304)

[6. Principais Dificuldades 10](#_Toc511227305)

[7. Contribuição no Projeto 10](#_Toc511227306)

[8. Conclusão 11](#_Toc511227307)

### Introdução

Na unidade curricular de Conceção e Análise de Algoritmos foi nos proposto a resolução de um problema onde seria implementado um sistema eficiente de gestão da rede de autoestradas de Portugal. Procedeu-se   
à implementação um sistema que determina os percursos alternativos que cada automóvel deve realizar caso exista um acidente grave que obrigue ao corte de um troço, como por exemplo um incêndio. Esta solução deveria ter em conta o cálculo do percurso mais curto de um ponto inicial para um final sendo que poderiam existir percursos cortados.

### Descrição do Problema

# 2.1 Input

Construção de um grafo G= (V,E), em que

* Vértices (V) – representam os vários pontos num mapa de estradas.
* Arestas (E) – representam as ligações entre os vários pontos do mapa assim como a capacidade máxima e o tráfego atual da via.
* Nó de início e nó de destino.

# 2.2 Introdução de dados

Ficheiros com o tráfego a percorrer as estradas.

# 2.3 Output

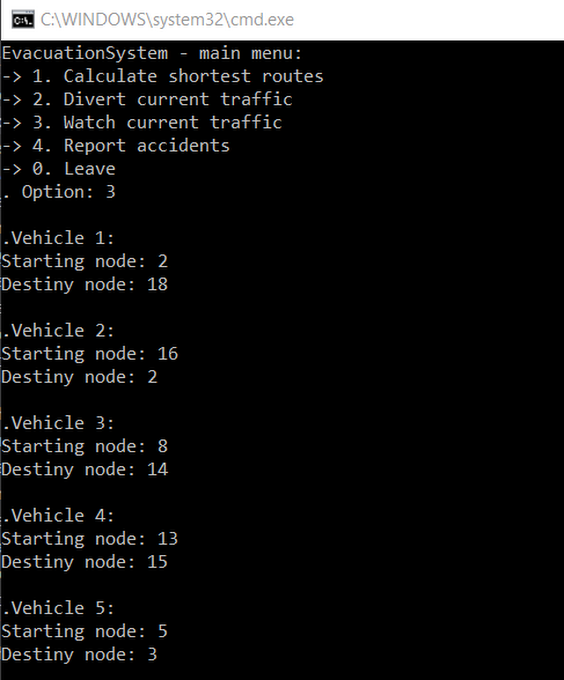
O percurso que deve ser realizado de um ponto inicial para um ponto final, ou todos os percursos possíveis a partir de um ponto tendo em conta a existência de algum troço fechado, apresentado o percurso realizado no menor tempo possível.

# 2.4 Objetivo

Apresentar o percurso que deve ser realizado para evitar um troço fechado e chegar ao destino em segurança, no menor tempo possível, processando todo o tráfego. O caminho considerado mais rápido, neste caso, como não se tem informação sobre as velocidades dos veículos, foi apenas a distância.

# 

# 2.5 Restrições

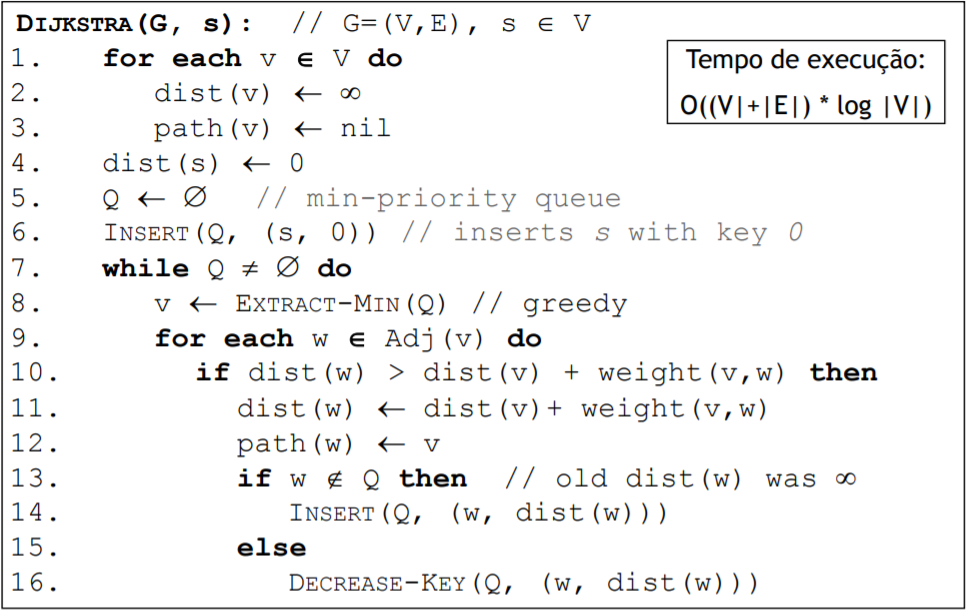
Evitar um troço fechado e cada troço da autoestrada possui uma capacidade limitada, isto é, número máximo de automóveis que suporta.

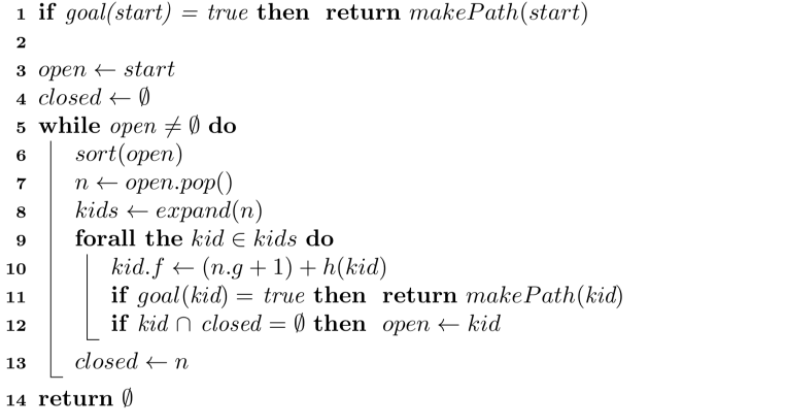
### Algoritmos implementados

Na realização do trabalho foram implementados dois algoritmos para encontrar o caminho mais curto: o de Dijkstra e o A\*.

O Algoritmo de Dijkstra, pelo facto de encontrar o percurso mais curto num grafo dirigido ou não dirigido em tempo **O((arestas+vértices]) \* log (vértices))** e pelo facto de o grafo representar um rede viária, ou seja, é esparso (|E| ~ |V|), foi considerado como uma das possíveis soluções para este problema.

No caso do algoritmo A\*, o tempo de execução para grafos com grande número de arestas por nó é menor que no algoritmo de Dijkstra. Se existirem poucas arestas por nó, o algoritmo A\* torna-se menos eficiente.



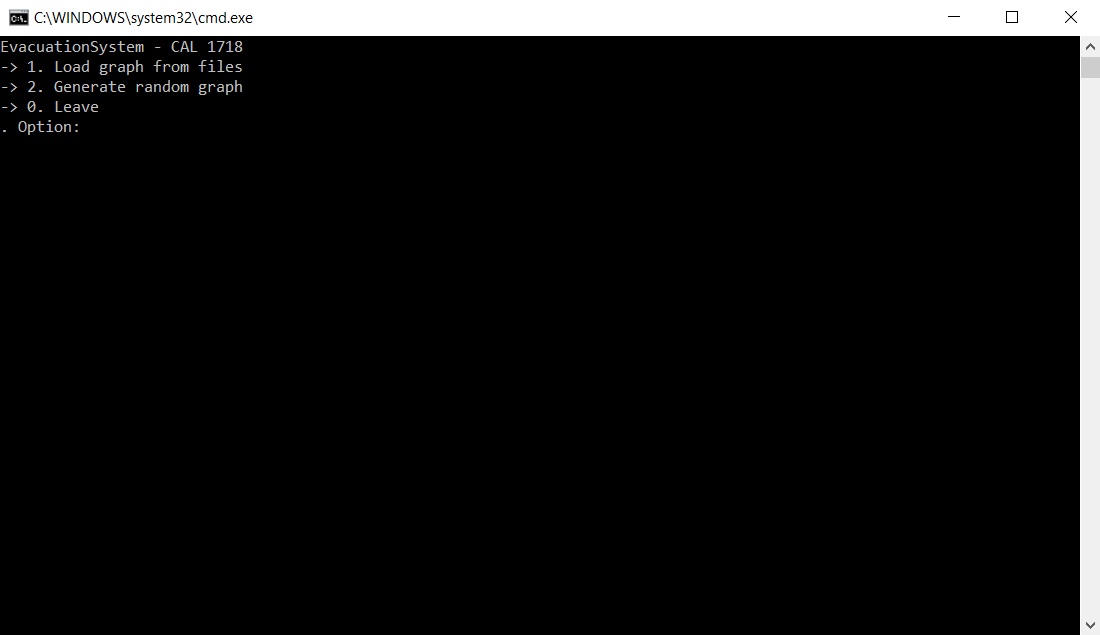


Pseudocódigo do Algoritmo A\*

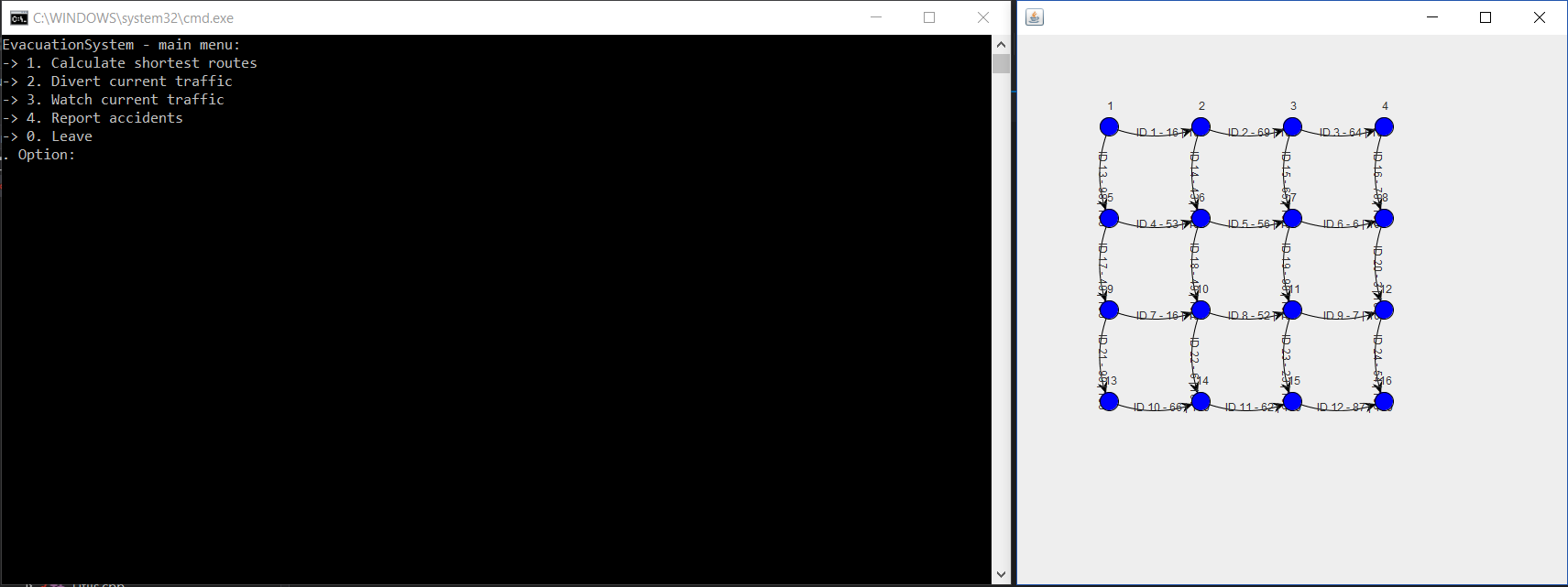
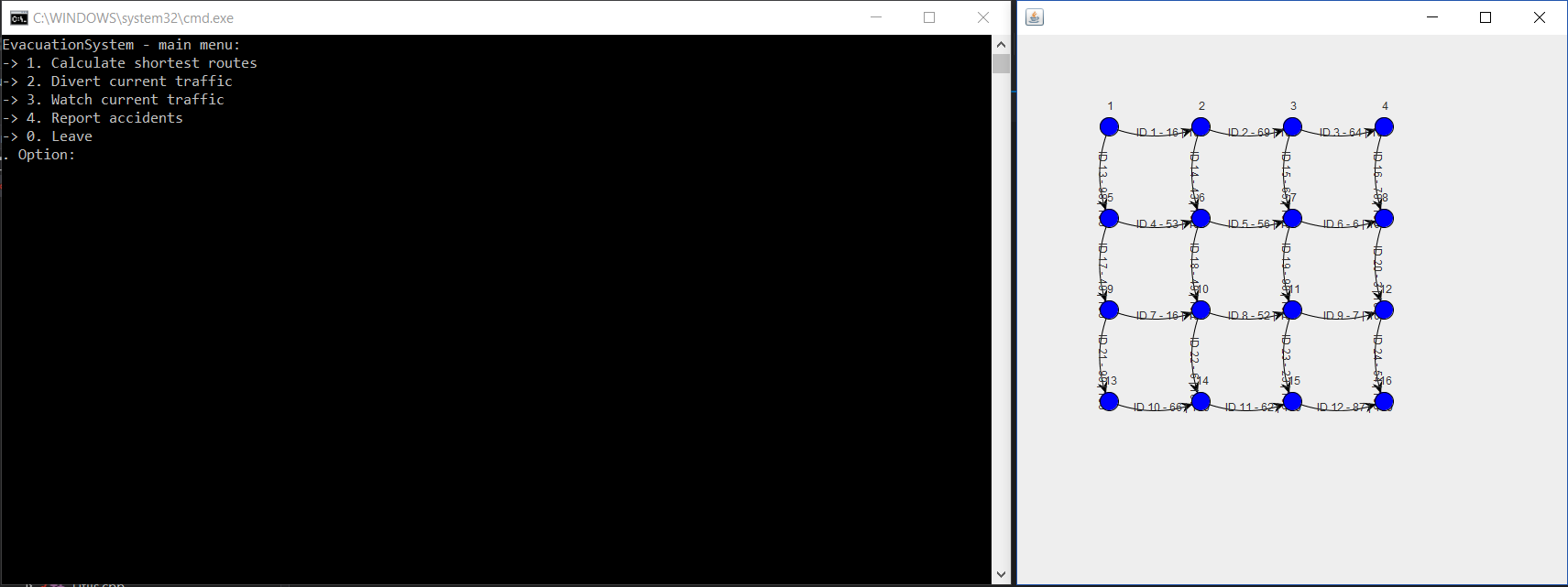
Pseudocódigo do Algoritmo de Dijkstra

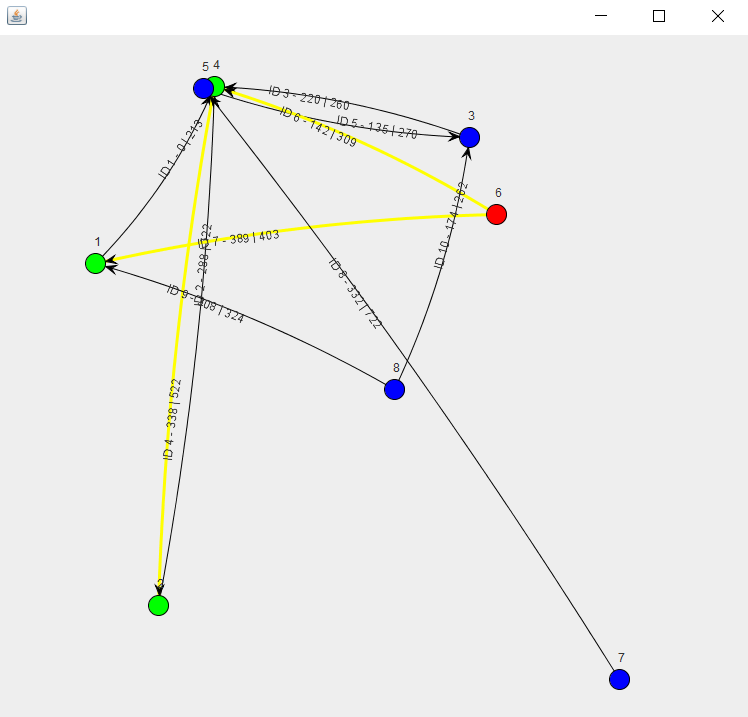
### Casos de Utilização

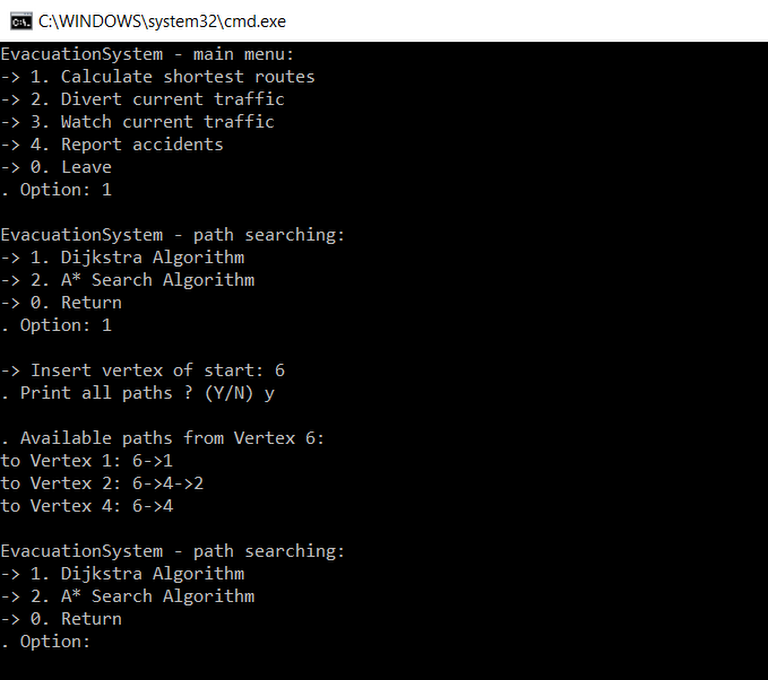
A interface do programa foi desenvolvida para a linha de comandos, sendo que todas as instruções são inseridas através desta. Todas os menus, incluindo inputs e outputs estão definidos na classe “Utils.cpp”.

Inicialmente opta-se por carregar um grafo já existente ou gerar um aleatoriamente.

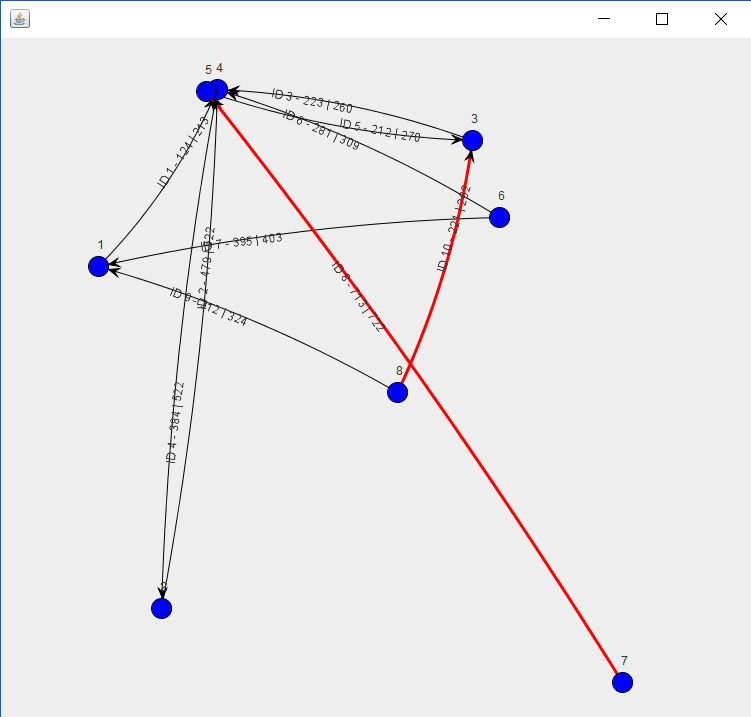
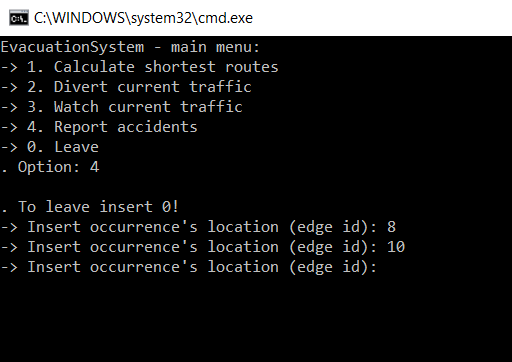
De seguida é possível selecionar uma das seguintes opções: calcular o caminho mais curto, divergir e observar o tráfego atual e reportar um novo acidente. É também apresentado o esquema do grafo, com os respetivos IDs de cada nó e aresta e com a sua capacidade máxima e atual.

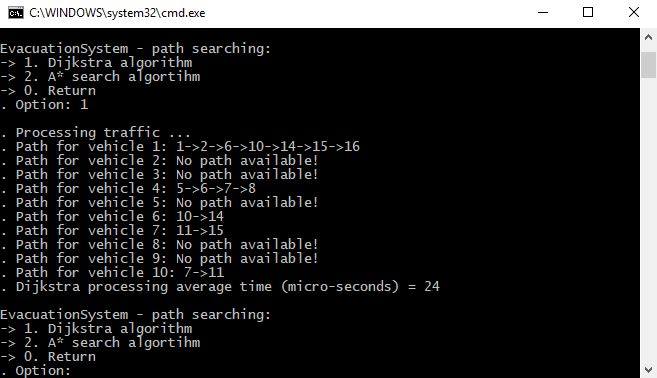


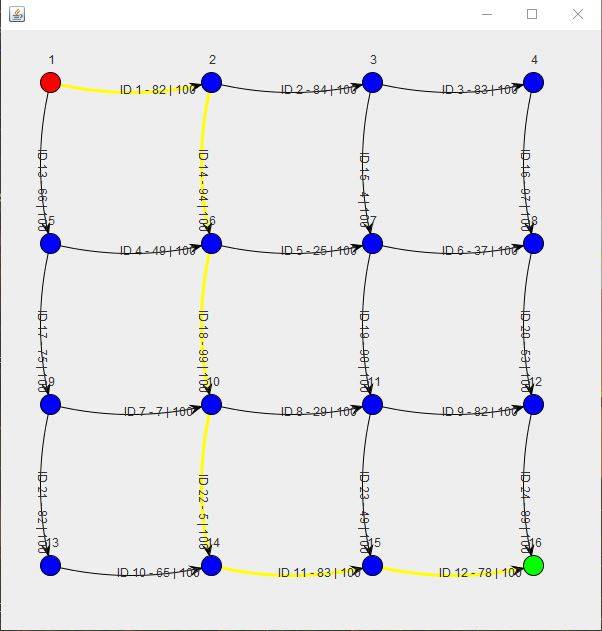
Na primeira opção é possível optar pelo algoritmo de Dijkstra e de A\*, seleciona-se o ID do nó inicial e opta-se por exibir todo os percursos possíveis a partir desse nó ou a inserção de um nó de destino específico. Graficamente, os percursos ficam representados na cor amarela, o nó de partida a vermelho e o(s) nó(s) de chegada a verde.

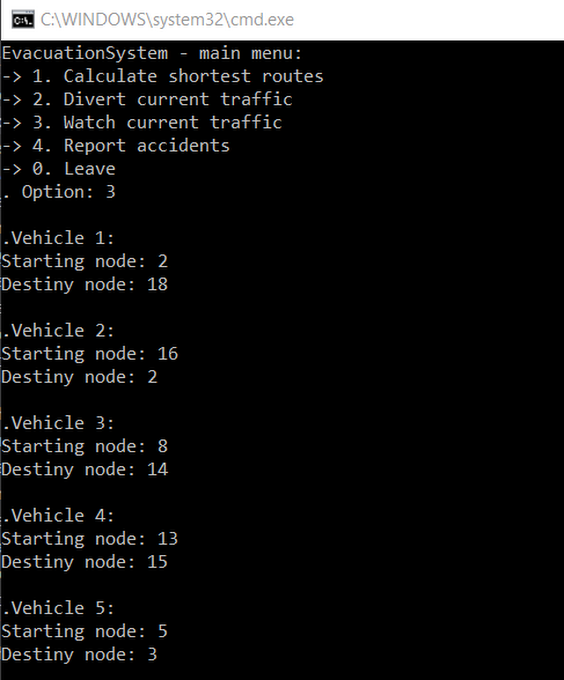


Na opção de reportar acidentes é possível marcar uma aresta, através do seu ID como inacessível, ficando esta marcada a vermelho.



Na opção 2 é possível realizar o desvio do trânsito, tendo em conta a existência ou não de desvios. É atualizado também, ao mesmo tempo, o número de veículos existentes em cada estrada.



Na opção 3 é possível ver todo o tráfego não processado, sendo este gerado aleatoriamente.

### Diagrama de Classes



### Principais Dificuldades

Na realização deste projeto consideramos que foram encontradas mais dificuldades na divisão de tarefas. Consideramos também que o enunciado fornecido era pouco específico e ambíguo em relação às especificações do trabalho a realizar.

### Contribuição no Projeto

Consideramos que a divisão de tarefas foi a seguinte: Luís Borges - 20% e Patrícia Janeiro - 10%, Sandro Campos - 70%.

### Conclusão

Com a realização deste projeto foi possível um maior contacto com algoritmos de pesquisa do caminho mais curto em grafos, conceitos indispensáveis no âmbito do curso de Engenharia Informática. Tentámos usar mapas reais na realização do trabalho mas tal não foi possível, sendo todos os mapas apresentados fictícios.